Beyond Pooling

Gordon V. Cormack
University of Waterloo
gvcormac@uwaterloo.ca

Maura R. Grossman
University of Waterloo
maura.grossman@uwaterloo.ca

ABSTRACT

Dynamic Sampling is a novel, non-uniform, statistical sampling strategy in which documents are selected for relevance assessment based on the results of prior assessments. Unlike static and dynamic pooling methods that are commonly used to compile relevance assessments for the creation of information retrieval test collections, Dynamic Sampling yields a statistical sample from which substantially unbiased estimates of effectiveness measures may be derived. In contrast to static sampling strategies, which make no use of relevance assessments, Dynamic Sampling is able to select documents from a much larger universe, yielding superior test collections for a given budget of relevance assessments. These assertions are supported by simulation studies using secondary data from the TREC 2017 Common Core Track.

ACM Reference Format:
Gordon V. Cormack and Maura R. Grossman. 2018. Beyond Pooling. In *SIGIR '18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, July 8–12, 2018, Ann Arbor, MI, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3209978.3210119

1 INTRODUCTION

This paper argues that the pooling method should be replaced by "Dynamic Sampling" ("DS"), in which active learning is used to select a stratified sample of documents for assessment to form the "gold standard" of relevance in an information retrieval ("IR") test collection. The essential idea is to adapt Scalable Continuous Active Learning ("S-CAL") [4]—a technology-assisted review method that repeatedly draws samples for assessment from strata of exponentially increasing size—for this task.

Each stratum consists of the documents deemed next-most-likely to be relevant by a learning algorithm, based on prior assessments. The feature representation employed by the learning algorithm may be derived from the relevance rankings afforded by the systems to be evaluated, relevance rankings afforded by reference systems, document content, or any combination of these approaches.

Together, the samples comprise a statistical sample of an arbitrarily large portion of the entire corpus—not just the top-ranked documents submitted for evaluation—whose size is bounded by a fixed assessment budget, regardless of the number of relevant documents in the corpus (R). An unbiased Horvitz-Thompson estimate of recall-independent measures, such as precision at rank ($P@k$), or R, can be derived from this sample [11]. Estimators for recall-dependent measures, such as average precision, R-precison ($P@R$), or NDCG, which employ a non-linear combination of these elementary estimators, entail bias, which can be mitigated by arranging a low-variance estimate of R.

Dynamic Sampling is particularly suitable for creating test collections that may be used to evaluate the effectiveness of methods that are not available at the time of construction. Provided enough relevant documents are found by DS to provide a low-variance estimate of R for each topic, future methods have no systematic advantage or disadvantage relative to those whose rankings were used as input to the DS process.

Using data from the TREC 2017 Common Core Track[1] [1], we simulate the use of DS to create test collections using assessment budgets of 100, 200, 300, and 600 documents per topic, which compare favorably to the official Common Core test collection, which was derived using the "max mean" dynamic-pooling strategy [9].

2 THE TREC 2017 COMMON CORE TRACK

The TREC 2017 Common Core Track ("Core Track") offered a strong baseline depth-100 subset pool consisting of 30,030 relevance assessments over 50 topics [9]. The authors employed the AutoTAR active learning method [5] to assess 11,825 documents for the same 50 topics. These two efforts yielded 9,002 and 8,986 positive assessments, respectively, but only 3,715 in common between them. For the purpose of this exposition, we consider ground truth to be the union of these two sets.

The Core Track baseline is atypical because of two factors that enured to its advantage. First, the topics had been previously used for the TREC 2004 Robust Track [16], yielding 76,783 relevance assessments, albeit for a different set of documents. Thirty-three of the topics had also been used for the TREC 2005 Robust Track [17], yielding 23,911 assessments for yet another set of documents. Of the 55 system rankings used to form the Core Track depth-100 pool, 14 were influenced by these 100,694 prior assessments.

A second factor enuring to the advantage of the baseline was the influence of manual assessments conducted by participating teams such as ourselves. At least one other team also employed active learning and assessed many thousands of documents in the course of their participation; a total of 12 system rankings (including three of ours) were influenced by manual effort. The remaining 29 rankings used to form the pool were derived using fully automatic methods, without influence from historical or current relevance assessments.

While 75 runs, each consisting of 10,000 documents per run, were submitted by participating teams, the Core Track assessment effort considered only the top-ranked 100 documents from 55 runs deemed "highest priority" by each of the Track participants.

[1] *See* https://trec-core.github.io/2017/.

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

ACM ISBN 978-1-4503-5657-2/18/07.
https://doi.org/10.1145/3209978.3210119

The experiments detailed below evaluate Dynamic Sampling methods that consider all submitted documents; a pool consisting of one-third more runs and one-hundred times the pool depth. We also consider the effect of restricting the pool to the 29 fully automatic methods. Finally, we consider DS methods that use no pool at all, selecting documents for assessment from the universe of documents, based on their *tf-idf* word representation.

3 DYNAMIC POOLING

Move-to-front pooling ("MTF") [6] is arguably the first application of active learning to the selection of documents for assessment and inclusion in an IR test collection [13]. Unlike the depth-k pooling method, which selects the top-ranked k documents from each of a set of participating runs for assessment, active learning methods repeatedly select documents from among those not yet assessed, based on the relevance assessments rendered for previously selected documents. A number of studies [9, 13] indicate that for $k < 100$, MTF and other active learning methods yield better test collections than depth-k pooling, for the same number of assessments, relative to a gold-standard test collection constructed using depth-100[2] pooling, which has been the de facto standard for nearly three decades.

The assumption that depth-100 pooling constitutes ground truth has influenced current practice, which typically applies active learning methods to select documents only from the depth-100 pool. This "dynamic pooling" approach requires fewer relevance assessments than the depth-100 pool, but yields an inferior test collection that can, at best, approach the quality of the depth-100 pool. Hereafter, we use the term "subset pooling" to refer to any strategy that selects a subset of a fixed-depth pool for assessment, and "dynamic pooling" to refer to a subset-pooling strategy that employs active learning. Thus, depth-k, meta-ranking, and statistical sampling are subset-pooling strategies, while MTF, hedge, and "bandit" methods [9] are dynamic-pooling strategies.

4 SUBSET SAMPLING

The literature describes several methods to employ statistical sampling as a subset-pooling strategy [3, 11, 19]. The general approach is to draw a uniform or stratified random sample from the pool for assessment, and then to measure the effectiveness of a run by applying a statistical estimator to approximate the value of some effectiveness measure (*e.g.*, P@k or average precision), were the entire pool to be judged.

The "infAP" family of methods [19] (generically, "infAP") use a separate statistical estimator for each stratum, and combine the results to form an overall estimate. The "statAP" family of methods [11] (generically, "statAP"), on the other hand, employ a Horvitz-Thompson estimator [10] to form an overall estimate based on the inclusion probabilities of the relevant documents in the strata, without regard to the strata from which the documents came. The minimal test collections family of methods [3] (generically, "MTC") are not statistical sampling methods, as they provide intentionally biased estimates for the purpose of distinguishing among runs.

[2]The arguments here apply to depth-K pooling where $K > k$, but depth-100 is nearly universal, and should be considered synecdoche for depth-K.

Dynamic Sampling differs from subset sampling in that it is not constrained to selecting documents from a pool, and it identifies strata sequentially in response to the relevance assessments for previous strata. Of the methods used for subset sampling, only statAP is amenable for DS, because it provides a reasonably unbiased estimate, even with many sparsely sampled strata. In our empirical work, we use Pavlu's reference implementation of statAP.[3]

5 TOTAL RECALL

Interactive Search and Judging ("ISJ"), which involves a sustained search effort to identify and label a substantial fraction of all relevant documents in a collection, has been shown to yield test collections of comparable quality to those yielded by depth-100 pooling, with less effort [6, 14]. More recently, a particular active learning approach has been shown to compare favorably to ISJ, while obviating the need for repeated query reformulation by a search expert [5]. The TREC 2002 Filtering Track [15] coordinators used a similar method to construct a substantially complete set of relevance assessments prior to the conduct of the Track, so that the relevance assessments could be used to simulate real-time feedback, as well as to evaluate the effectiveness of the submitted runs. Subsequent application of depth-100 pooling uncovered a number of relevant documents that were not discovered by the prior effort, but those additional documents were found to have an insubstantial impact on the resulting effectiveness estimates.

The current state of the art in active learning for this purpose is AutoTAR [5], as realized by the TREC Total Recall Track Baseline Model Implementation ("BMI").[4] AutoTAR was used to label test collections prior to the TREC 2015 and 2016 Total Recall Tracks, and also was provided to Track partcicipants as a baseline method. No submitted run consistently bettered BMI, whether the effectiveness was measured using the official AutoTAR-selected assessments, statistically sampled assessments, or post-hoc assessments by one of the participating teams [7].

The authors used a modified version of BMI to prepare their submission to the Core Track [2]. Although at the time of this writing, only 50 of the Core Track topics had been assessed, 250 topics were provided to participants. For 250 topics, the authors spent 64.1 hours assessing 42,587 documents (on average, 15.4 mins/topic; 5.4 secs/doc), judging 30,124 of them to be relevant (70.7%). Of these judgments, 11,825 pertained to the 50 topics that have, to date, been officially assessed. Given the Core Track's tight timeline, our efforts were necessarily incomplete. Nonetheless, we judged relevant roughly the same number of documents as the Core Track's official assessors. Of these, more than one-third were not in the Core Track's assessment pool. A statistical sample embedded in one of our submissions indicates that the majority of these unassessed documents would have been judged relevant, had they been assessed [2].

The TREC 2015 and 2016 Total Recall results show that AutoTAR can achieve near-perfect recall given $2R + 100$ assessments per topic, where R is the number of relevant documents to be found [8, 12]. According to this rule of thumb, we would have had to expend

[3] *See* http://trec.nist.gov/data/million.query/07/statAP_MQ_eval_v3.pl. This implementation corrects a serious error in an unpublished but commonly cited manuscript [10]. The description in Pavlu's dissertation [11] is correct.

[4] *See* http://cormack.uwaterloo.ca/trecvm/.

about $2\frac{1}{2}$ times as much assessment effort to achieve near-perfect recall, if we assume that there were 14,273 relevant documents, in accordance with our ground-truth assumption.

Over and above the absolute cost of obtaining relevance assessments is the problem of budgeting and resource allocation. It is not known in advance how many documents will be relevant to each topic, and therefore how assessment resources should be apportioned or scheduled over topics. The variance of R in the Core Track topics is huge, ranging from 9 to 1,377, with an average of 285 and a mean of 191. Topics with large R consume an inordinate fraction of the budget, as may topics with very small R, where it could be a challenge to find *any* relevant documents.

6 DYNAMIC SAMPLING

Dynamic Sampling adapts AutoTAR to identify a sequence of strata with exponentially increasing size, which are sampled with diminishing frequency, so that the total number of documents assessed per topic is equal to a sampling budget that is fixed in advance. A sampling budget of precisely 600 assessments per topic, or 30,000 assessments in total, nearly equals the 30,030 assessments of the Core Track depth-100 subset pool.

The exponentially increasing strata are precisely the exponentially growing batches of documents identified by AutoTAR; however, a uniform random sample of each batch is selected for assessment, and used to train the learning method, which then identifies the next batch. The main idea is derived from Cormack and Grossman's "S-CAL" [4], which is itself derived from AutoTAR. While the purpose of S-CAL is to achieve the best possible classifier over an infinite population, with statistical estimation playing a supporting role, the purpose of Dynamic Sampling is to achieve the best possible statistical estimator, with classification playing a supporting role. In response to this difference in emphasis, we amended the procedure by which the sampling rate decays.

Algorithm 1 Dynamic Sampling Algorithm (from S-CAL [4]).

1: Construct a relevant pseudo-document from topic description.
2: The initial training set contains only the pseudo-document.
3: Set the initial batch size B to 1.
4: Set the initial decay threshold T to hyper-parameter N.
5: Temporarily augment the training set by adding 100 random documents from the collection, labeled "not relevant."
6: Score all documents using a model induced from training set.
7: Remove the random documents added in step 5.
8: Select the highest-scoring B documents not previously selected.
9: Draw $n = \lceil \frac{B \cdot N}{T} \rceil \leq B$ random documents from step 8.
10: Render relevance assessments for the n documents.
11: Add the assessed documents to the training set.
12: Increase B by $\lceil \frac{B}{10} \rceil$.
13: If the number of assessed relevant documents $R \geq T$, double T.
14: Repeat 5 through 13 until assessment budget A is reached.

The Dynamic Sampling method is outlined in Algorithm 1. The only input to the method, other than the document collection, topic statement, and assessment budget A, is a hyper-parameter N controlling the decay of the sampling rate. At the outset, and until at least N relevant documents are discovered, every document in every batch presented by AutoTAR is presented for assessment, and used to train the model. In the event that N documents are never found, documents are examined exhaustively until the assessment budget is met. Once N relevant documents are found, the sampling rate is halved until N more relevant documents are found, and so on, until the assessment budget is reached.

N quantifies a tradeoff between the objectives of selecting the largest possible number of relevant documents for assessment, and sampling a universe containing the largest possible number of relevant documents. In the extreme case where $N = A$, and also in the extreme case where $R \leq N$, DS is exactly AutoTAR. In the opposite extreme where $N \ll R$, the sampling frequency will nearly vanish, and the universe will approach the entire document population. We evaluated all twenty combinations of $A \in \{100, 200, 300, 600\}$ and $N \in \{12, 25, 50, 100, 200\}$.

7 FEATURE ENGINEERING

In one version of DS, we represented each document as a *tf-idf* word vector, exactly as calculated by BMI. In a second version, we represented each document as a reciprocal-rank vector with d dimensions, corresponding to d runs used to form a depth-10000 pool. For the full pool consisting of all Core Track submissions, $d = 75$; for the automatic pool consisting exclusively of automatic runs, $d = 29$. The value of each feature is set to $\frac{1}{d} \cdot \frac{1}{50+\rho}$, where ρ is the rank of the document according to the corresponding run. In Table 1, rows labeled "DN" denote results using *tf-idf* and hyper-parameter N; rows labeled "RN" denote results using rank features from the full pool; and rows labeled "AN" denote results using the automatic pool. Rows labeled "RDN" and "ADN" denote results that average the scores from two separate models in Algorithm 1, step 6: one using *tf-idf* features; the other using rank features from the full pool or automatic pool, respectively.

8 RESULTS

The quality of a test collection may be characterized by how well it computes effectiveness measures for particular runs, or by how well it ranks the relative effectiveness of the various runs that it may be called upon to evaluate. For this short paper, we focus on how well a test collection ranks the 75 Core Track runs according to MAP, using Kendall's τ to compare rankings. We have also computed τ_{AP} [18], variance, and bias, which show the same effect. Table 1 shows τ for three subsets of the Core Track pool, and the variants of our Dynamic Sampling method, sorted by the column labeled $A = 600$.

"Core" denotes the 30,030-document pool identified by the Core Track. "Core-Auto" denotes a subset of Core containing assessments only for documents that are among depth-100 pool formed using only the 29 automatic rankings. Core-Auto contains 15,024 assessments—about 300 per topic—and is therefore included under $A = 300$ in Table 1. "Core-Sample" denotes a two-stratum sample, drawn by the Core Track organizers, of the depth-75 pool of the 55 runs used to form the Core Track pool. The first stratum consists of all documents in the top-10 pool, while the second stratum consists of a 20% sample of all documents in the top-75 pool, excluding the documents in the first stratum. Core-Sample contains 15,024 assessments—also about 300 per topic—and is included under $A = 300$.

Table 1: Kendall τ correlation between ground truth and test collections built using pooling and dynamic sampling.

Method	*Assessment Budget Per Topic**			
	A=100	*A=200*	*A=300*	*A=600*
AD50	.835	.932	**.971**	**.986**
RD25	.881	**.956**	.969	.982
R25	.734	.936	.970	.982
R50	.531	.887	.941	.981
R100	.431	.843	.914	.980
RD100	.813	.908	.953	.978
RD50	.837	.933	.963	.977
AD200	.814	.895	.940	.977
A25	.735	.912	.951	.975
RD200	.813	.897	.940	.974
AD100	.815	.918	.945	.972
Core				.971†
R12	.873	.947	.962	.967
D50	.794	.918	.956	.967
RD12	**.930**	.939	.956	.966
D12	.873	.943	.966	.965
AD12	.892	.954	.962	.964
D100	.802	.890	.931	.962
AD25	.882	.946	.949	.961
Core-Sample			.944†	
D200	.796	.886	.926	.960
A100	.458	.807	.888	.960
D25	.838	.925	.958	.959
R200	.436	.794	.881	.958
A50	.550	.853	.918	.957
A12	.839	.921	.942	.955
A200	.450	.775	.850	.941
Core-Auto			.420†	

Key

Core	Core Track assessment pool
Core-Sample	Core Track depth-75 stratified sample
Core-Auto	Core Track pool, automatic only
R*N*	Dynamic, all 75 rankings
A*N*	Dynamic, 29 automatic rankings only
D*N*	Dynamic, content
RD*N*	Dynamic, all 75 rankings + content
AD*N*	Dynamic, 29 automatic rankings + content
N	denotes hyper-parameter *N* (see text)
*	Fixed budget per topic unless indicated by †
†	Variable budget per topic

9 CONCLUSIONS

For assessment budget $A = 600$, the Core Track collection is near the median of the various Dynamic Sampling methods. Remarkably, the top-scoring system for this budget is AD50, which uses only automatic rankings and document content—along with relevance feedback—to select documents for assessment. While we cannot say that the difference between AD50 and RD50 results is substantial, it does appear that RD*N* offers no material advantage over AD*N*. It also appears that AD*N* and RD*N* compare favorably with Core, except for the smallest $N = 12$. Of the single-model methods, R*N* fares the best, and offers an apparently superior plug-in replacement to the dynamic-pooling method used by the Core Track. The results for D*N* appear to be slightly inferior to those for Core; however, it should be noted that D*N* uses no pooling whatsoever, and is the only technique presented here that could be used to construct a test collection prior to the conduct of an evaluation task. A*N* is remarkably good, considering the narrow set of rankings from which it is derived, but inferior to the other methods.

For $A = 300$, the results for AD*N*, RD*N*, and R*N* are nearly as good as for $A = 600$, and better than Core-Sample. Perhaps unsurprisingly, the results for Core-Auto are abysmal, confirming received wisdom that the pooling method depends on the presence of manual runs. The extreme difference between between Core-Auto and AD50 is noteworthy because both methods rely on information from the same set of automatic runs, and use the same number of assessments.

The results for $A = 100$ and $A = 200$ show an overall degradation and an increase in variability, but may still reflect useful methods, as they require one-sixth to one-third as many assessments as the Core Track effort. An open question remains: Would assessing all 250 topics with DS and a budget of $A = 120$ have yielded a better test collection, for the same total effort of 30,000 assessments? We believe so.

REFERENCES

[1] James Allan, Evangelos Kanoulas, Donna Harman, and Ellen Voorhees. TREC 2017 Common Core Track overview. In *TREC 2017*.
[2] Anonymized. Participation in the TREC 2017 Core Track. In *TREC 2017*.
[3] Ben Carterette, James Allan, and Ramesh Sitaraman. Minimal test collections for retrieval evaluation. In *SIGIR 2006*.
[4] Gordon V Cormack and Maura R Grossman. Scalability of continuous active learning for reliable high-recall text classification. In *CIKM 2016*.
[5] Gordon V Cormack and Maura R Grossman. Autonomy and reliability of continuous active learning for technology-assisted review. *arXiv:1504.06868*, 2015.
[6] Gordon V. Cormack, Christopher R. Palmer, and Charles L. A. Clarke. Efficient construction of large test collections. In *SIGIR 1998*.
[7] Maura R Grossman, Gordon V Cormack, and Adam Roegiest. Automatic and semi-automatic document selection for technology-assisted review. In *SIGIR 2017*.
[8] Maura R Grossman, Gordon V Cormack, and Adam Roegiest. TREC 2016 Total Recall Track Overview. In *TREC 2016*.
[9] David E Losada, Javier Parapar, and Álvaro Barreiro. Feeling lucky?: Multi-armed bandits for ordering judgements in pooling-based evaluation. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pages 1027–1034. ACM, 2016.
[10] V Pavlu and J Aslam. A practical sampling strategy for efficient retrieval evaluation. *College of Computer and Information Science, Northeastern University*, 2007.
[11] Virgil Pavlu. *Large Scale IR Evaluation*. Northeastern University, 2008.
[12] Adam Roegiest, Gordon V Cormack, Maura R Grossman, and Charles L A Clarke. TREC 2015 Total Recall Track Overview. In *TREC 2015*.
[13] Mark Sanderson et al. Test collection based evaluation of information retrieval systems. *Foundations and Trends in Information Retrieval*, 4(4):247–375, 2010.
[14] Mark Sanderson and Hideo Joho. Forming test collections with no system pooling. In *SIGIR 2004*.
[15] Ian Soboroff and Stephen Robertson. Building a filtering test collection for TREC 2002. In *SIGIR 2003*.
[16] Ellen M. Voorhees. Overview of the TREC 2004 Robust Track. In *Proceedings of the 13th Text REtrieval Conference*, Gaithersburg, Maryland, 2004.
[17] Ellen M Voorhees. The TREC 2005 Robust Track. In *ACM SIGIR Forum*, volume 40. ACM, 2006.
[18] Emine Yilmaz, Javed A Aslam, and Stephen Robertson. A new rank correlation coefficient for information retrieval. In *SIGIR 2008*.
[19] Emine Yilmaz, Evangelos Kanoulas, and Javed A. Aslam. A simple and efficient sampling method for estimating AP and NDCG. In *SIGIR 2008*.